
# Scaled Heavy-Ball Acceleration of the Richardson-Lucy Algorithm for 3D Microscopy Image Restoration

Wang, H., & Miller, P. C. (2014). Scaled Heavy-Ball Acceleration of the Richardson-Lucy Algorithm for 3D Microscopy Image Restoration. *IEEE Trans. on Image Processing*, *23*(2), 848-854. https://doi.org/10.1109/TIP.2013.2291324

**Published in:**
IEEE Trans. on Image Processing

**Document Version:**
Peer reviewed version

**Queen's University Belfast - Research Portal:**
Link to publication record in Queen's University Belfast Research Portal

**Publisher rights**
© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

**General rights**
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Download date:04. May. 2023

# Scaled Heavy-Ball Acceleration of the Richardson-Lucy Algorithm for 3D Microscopy Image Restoration

Hongbin Wang, Paul Miller

***Abstract*—The Richardson-Lucy algorithm is one of the most important in image deconvolution. However, a drawback is its slow convergence. A significant acceleration was obtained by the technique proposed by Biggs and Andrews (BA), which is implemented in the *deconvlucy* function of the *Image Processing* MATLAB toolbox. The BA method was developed heuristically with no proof of convergence. In this paper, we introduce the Heavy-Ball (H-B) method for Poisson data optimization and extend it to a scaled H-B method, which includes the BA method as a special case. The method has a proof of the convergence rate of $O(k^{-2})$, where *k* is the number of iterations. We demonstrate the superior convergence performance, by a speedup factor of five, of the scaled H-B method on both synthetic and real 3D images.**

***Index Terms*—Deconvolution, Poisson noise, Richardson-Lucy algorithm, Heavy-ball acceleration**

## I. INTRODUCTION

Restoration of 3D confocal microscopy images to remove blur and noise is an important preprocessing step for automated analysis and also for visualization by an end-user. One characteristic of 3D confocal microscopy imagery is that it is degraded mainly by Poisson noise under low light conditions. Perhaps the best well-known of restoration techniques for Poisson data is the Richardson-Lucy (RL) algorithm [1, 2].

### *A. Problem Formulation*

Given an object $x$ and its image $y$, which is acquired through an imaging system with a point-spread function (PSF) $K$ and corrupted by a Poisson noise process, then the image formation model is given by

$$y \sim Poisson(Kx + b), \tag{1}$$

where *x* and *y* are represented as *n*-element vectors, and $n$ is the total number of voxels or pixels in the image. The PSF function *K* is an $n \times n$ matrix. The background term *b* is a *n*-element vector. It stands for background emission [3] and can be estimated by preprocessing of the raw image *y* [4]. From a Bayesian viewpoint, the conditional probability of image *y* given object *x* is as follows

$$p(y \mid x) = \prod_{i=1}^{n} \frac{(e_i^T Kx + b)^{y_i}}{y_i!} e^{-(e_i^T Kx + b)}, \tag{2}$$

where $e_i$ is the $i^{\text{th}}$ canonical basis unit vector. Taking $C(x) = -\log p(y|x)$ as a cost function, then

$$C(x) = x - \sum_{i=1}^{n} \log\left(e_i^T Kx + b\right), \tag{3}$$

where we have neglected the constant $\sum_{i=1}^{n} \log(y_i!)$ and incorporated the PSF normalization, i.e., $\sum_{i=1}^{n} e_i^T K = 1$. The gradient of $C(x)$ is

$$\nabla C(x) = \mathbf{1} - \sum_{i=1}^{n} \frac{y_i}{e_i^T Kx + b} K^T e_i, \tag{4}$$

where $\mathbf{1}$ is the unit vector. And minimizing $C(x)$ w.r.t *x*, gives the RL iterative equation

$$x^{k+1} = \left(x^k\right)^T \sum_{i=1}^{n} \frac{y_i}{e_i^T Kx^k + b} K^T e_i, \tag{5}$$

where $x^k$ is the $k^{\text{th}}$ iteration of *x*.

### *B. Gradient projection and related work*

We begin by considering Eqn. (5) as a special case of the gradient projection method [3, 9]. Given the same cost function used for RL, the general iteration equation of the gradient projection method is given by

$$x^{k+1} = P_\Omega\left[x^k - \left(\alpha^k\right)^T \nabla C(x)\right], \tag{6}$$

where $P_\Omega$ denotes projection onto $\Omega$, the feasible set of *x*, and $\alpha^k$ is an *n*-element vector denoting the $k^{\text{th}}$ step size. The RL algorithm can be interpreted as follows; $P_\Omega$ is the

nonnegative constraint, inserting the gradient of the cost function, Eqn.(4), and setting $\alpha^k = x^k$, gives Eqn.(5).

However, although the RL algorithm is very popular, it has drawbacks in terms of the restoration quality and implementation time for 3D images. To address the slow convergence of the RL algorithm many acceleration methods have been proposed, which try to select step size $\alpha^k$ [5], or adjust the search direction $\nabla C\left(x^k\right)$ [6, 7]. Unfortunately, in many cases, a reduction in the number of iterations is achieved only at the expense of increased computational cost per iteration, resulting in an insignificant speed-up [3].

There are two exceptional pieces of work that have achieved significant acceleration for the RL algorithm. One of these methods, based on extrapolation of the iterative point, was proposed by Biggs and Andrews (BA) in [8]. During each iteration of the BA technique, a predicted point $p^k$, instead of $x^k$, is used to calculate $x^{k+1}$ in Eqn.(5), where

$$p^k = x^k + \left(\beta^k\right)^T h^k, \tag{7}$$

and $h^k = x^k - x^{k-1}$. $\beta^k$ is the acceleration parameter, which can be calculated as

$$\beta^k = \frac{\sum_{i=1}^{n}\left(g^{k-1}\right)^T g^{k-2}}{\sum_{i=1}^{n}\left(g^{k-2}\right)^T g^{k-2}}, \tag{8}$$

where $g^k = x^{k+1} - p^k$ and $0 < \beta^k < 1$. A shortcoming of BA is that no convergence proof is available.

In the second of these works, another method, called scaled gradient projection, was proposed by Bonettini in [9]. It achieved similar performance to the BA method and has a convergence proof. However, it does not have a convergence rate proof.

A number of acceleration algorithms have been recently proposed that consider Gaussian noise instead of Poisson in the image formation model [10-13]. For example, the acceleration algorithm proposed by Nesterov [10], achieved a fast convergence rate of $O\left(k^{-2}\right)$, and was then extended to problems involving a convex non-smooth regularizer by the fast iterative shrinkage/thresholding algorithm (FISTA) [12]. Unfortunately, we cannot directly apply Nesterov's acceleration algorithm, or FISTA, here, as its attractive convergence rate is based on the fact that the negative Gaussian log-likelihood has a Lipschitz-continuous gradient, which is not the case for the negative Poisson log-likelihood.

Previously [14], we related the BA method to an optimization technique, called the Heavy-Ball (H-B) method [15], and presented some preliminary results. In this work, we present a rate of convergence proof for a new technique, called the scaled H-B method, which includes the BA method as a special case. By using the theoretical proof that the negative Poisson log-likelihood has a Lipschitz-continuous gradient if the variable value is positive, and replacing the Lipschitz constant with a positive definite matrix [9, 13], we prove that the scaled H-B method does not just converge, but that it has a faster convergence rate, of $O\left(k^{-2}\right)$, than that of the original RL algorithm, of $O\left(k^{-1}\right)$. We also empirically evaluate the scaled H-B method through experiments with simulated and real image data.

## III. HEAVY-BALL METHOD

In this section, we firstly introduce the H-B method, and then we propose a generalized H-B, called scaled H-B, and derive the BA method as a special case. Finally, we discuss the selection of step size and acceleration parameter.

### *A. Heavy-Ball method for Poisson data*

The H-B method is a general acceleration method for gradient projection. It was proposed by Polyak [15], and the basic idea is to enhance the iterative updating by adding a momentum term from the previous step. The method can be separated into an extrapolation step, given by Eqn. (7), and a gradient projection step

$$x^{k+1} = P_\Omega\left[p^k - \left(\alpha^k\right)^T \nabla C\left(p^k\right)\right]. \tag{9}$$

Normally, the gradient based method can only achieve a convergence rate of $O\left(k^{-1}\right)$ [16]. However, it can be proved that the H-B method achieves a best convergence rate of $O\left(k^{-2}\right)$ for the gradient based method [16] assuming:

- the cost function $C(x)$ is convex and differentiable, and $x \in X$, where $X$ is a closed convex set,
- $C(x)$ has a Lipschitz continuous gradient, $\left\|\nabla C\left(x^1\right) - \nabla C\left(x^2\right)\right\| \le L\left\|x^1 - x^2\right\|, \forall x^1, x^2 \in X$, where $L$ is the Lipschitz constant,
- $\alpha^k = \dfrac{\mathbf{1}}{L}$ and $\beta^k = \dfrac{\theta^k\left(1-\theta^{k-1}\right)}{\theta^k}\mathbf{1}$, where the sequence $\left\{\theta^k\right\}$ satisfies $\theta^0 = \theta^1 \in (0,1]$, $\left(1-\theta^{k+1}\right)\Big/\left(\theta^{k+1}\right)^2 \le 1\Big/\left(\theta^k\right)^2$, and $\theta^k \le 2/k+2$.

One possible choice for the acceleration parameter is

$$\beta^k = \frac{k-1}{k+2}\mathbf{1},\ \theta^k = \frac{2}{k+2} \tag{10}$$

Provided these are valid then we have [16]:

**Theorem 1**. *Let* $\{x^k\}$ *be a sequence generated by the H-B method, described by Eqns.* (7) *and (9), and* $x^*$ *be any optimal solution, then*

$$C(x^k) - C(x^*) \le \frac{2L}{(k+1)^2}\|x^0 - x^*\|, \forall k \ge 1, \qquad (11)$$

*where* $x^0$ *is the initial estimate.*

*Proof:* Our problem definition has to fulfill the above assumptions. Regarding the first assumption, the Poisson negative log-likelihood function is obviously convex and differentiable. By selecting appropriate values for $\alpha^k$ and $\beta^k$ the third assumption can be met. Generally, however, the Poisson negative log-likelihood function does not have a Lipschitz continuous gradient because of the logarithm. Several previous works have tried to attack this problem, either through proposing new methods, such as the alternating direction optimization [17], or through transforming the Poisson distribution to a Gaussian [18]. Fortunately, in microscopy images, there always exists a positive background $b$ [3], hence the $e_i^T Kx + b$ term in brackets in Eqn. (3) is positive. ■

Therefore, we have the following lemma to satisfy the second assumption.

**Lemma 1.** *The negative Poisson log-likelihood has a Lipschitz-continuous gradient if* $e_i^T Kx + b$ *is positive.* (*Proof is omitted as a similar result can be found in [19].)*

*B. Scaled Heavy-Ball and justification of BA method*

To justify the BA method, we propose a general form of H-B, called scaled H-B, which replaces the Lipschitz constant with a positive definite matrix to set the step size $\alpha^k = x^k$, thereby achieving better convergence. A similar idea has been explored in previous work [9, 13], however, we adapt it here to generalize the H-B method. The logic is that the Lipschitz gradient condition (second assumption) is required to guarantee the following inequality [16]

$$C(x^1) \le C(x^2) + (x^1 - x^2)^T \nabla C(x^2) + \frac{L}{2}\|x^1 - x^2\|^2. \qquad (12)$$

And that the above inequality is key to the proof of Theorem 1, which in turn leads to setting the step size $\alpha^k = \frac{\mathbf{1}}{L}$. However, $L$ is a loose estimate. If we can find a smaller Lipschitz constant, then, according to inequality (12), we will obtain faster convergence. This can be achieved through setting $\alpha^k = \beta^k$ and introducing a corresponding positive definite matrix $D = diag(x^k)$ to replace the Lipschitz constant. Hence, inequality (12) generalizes to

$$C(x^{k+1}) \le C(x^k) + (x^{k+1} - x^k)^T \nabla C(x^k) + \frac{1}{2}\|x^{k+1} - x^k\|_D^2, \qquad (13)$$

where $\|\bullet\|_D$ is the weighted norm associated with the positive definite matrix $D: \|x\|_D = \sqrt{x^T Dx}$. To prove inequality(13), we firstly need the following lemma:

**Lemma 2.** *Setting the step size* $\alpha^k = x^k$ *is an approximation of the inverse Hessian.*

*Proof:* Let $H$ be the Hessian of the cost function $C(x)$. Then, the inverse of the hessian is

$$H^{-1} = (K^T)^{-1} \sum_{i=1}^{n} \frac{(e_i^T Kx^k + b)(e_i^T Kx^{k+1} + b)}{y_i} e_i e_i^T K^{-1}.$$

From Eqn. (6), substituting for $x^{k+1}$ gives

$$H^{-1} = (K^T)^{-1} \left[ \sum_{i=1}^{n} \left\{ e_i^T K \left[ x^k - \alpha \left( \mathbf{1} - \sum_{j=1}^{n} \frac{y_j}{e_j^T Kx^k + b} K^T e_j \right) \right] + b \right\} \times \frac{(e_i^T Kx^k + b)}{y_i} e_i e_i^T \right] K^{-1}$$

Setting $H^{-1} = \alpha$ and rearranging gives

$$K^T \alpha K = \sum_{i=1}^{n} \left[ (e_i^T Kx^k)^2 + \left( e_i^T K\alpha \sum_{j=1}^{n} \frac{y_j}{e_j^T Kx^k + b} K^T e_j \right)(e_i^T Kx^k + b) - e_i^T K\alpha e_i^T Kx^k + 2e_i^T Kx^k b - e_i^T K\alpha b + b^2 \right] \frac{e_i e_i^T}{yi}$$

The summation over *j* only ever takes a value when *i* = *j* to give

$$K^T \alpha K = \sum_{i=1}^{n} \left[ (e_i^T Kx^k)^2 - e_i^T K\alpha (e_i^T Kx^k + b - y_i K^T e_i) + 2be_i^T Kx^k + b^2 \right] \frac{e_i e_i^T}{yi}.$$

Rearranging once more gives

$$e_i^T K\alpha K^T e_i y_i = (e_i^T Kx^k)^2 - e_i^T K\alpha (e_i^T Kx^k + b - y_i K^T e_i) + 2be_i^T Kx^k + b^2.$$

Finally, we have

$$e_i^T K\alpha (e_i^T Kx^k + b) = (e_i^T Kx^k + b)^2$$
$$\alpha = x^k + (e_i^T K)^{-1} b,$$

which completes the proof. ■

Since the Lipschitz constant $L$ is chosen as the largest eigenvalue of Hessian [19], the following inequality holds

$$\alpha^k = x^k \geq \frac{\mathbf{1}}{L} \tag{14}$$

In a manner similar to [13], we can replace inequality (12) in the proof of **Theorem 1** in [16], with inequality (13) to get the following theorem for the scaled H-B method.

**Theorem 2.** *Let* $\{x^k\}$ *be a sequence generated by the scaled H-B method, and* $x^*$ *be any optimal solution, then*

$$C(x^k) - C(x^*) \leq \frac{2}{(k+1)^2}\|x^0 - x^*\|_D^2, \forall k \geq 1 . \tag{15}$$

**Theorem 3.** *Scaled H-B converges faster than H-B.*

*Proof:* **Theorems 1** and **2** imply a convergence rate of $O(k^{-2})$, but $\|x\|_D \leq L\|x\|$. Therefore, scaled H-B converges faster. ■

We now provide a theoretical basis for the BA method, in that we can say it is a partially scaled H-B method. This is because it only replaces $x^k$ with $p^k$ inside the gradient as follows

$$x^{k+1} = P_\Omega\left[x^k - \alpha^k \nabla C(p^k)\right]. \tag{16}$$

Note that the BA method also has its special acceleration parameter defined by Eqn.(8), however, it is not easy to prove that $\beta^{BA} \leq \frac{k-1}{k+2}$. We can ensure the inequality by modifying $\beta^{BA}$ as follows

$$\beta^{BA} = \min\left\{\frac{\sum_{i=1}^{n} g^{k-1}g^{k-2}}{\sum_{i=1}^{n} g^{k-2}g^{k-2}}, \frac{k-1}{k+2}\right\}. \tag{17}$$

Thus, we know that **Theorem 2** holds for the BA method.

### *C. Choosing step size and acceleration parameter*

The two parameters in the scaled H-B method are step size $\alpha^k$ and acceleration parameter $\beta^k$. In the original H-B method [16], $\alpha^k$ is either set to $1/L$, which is seldom known in practice, or it is estimated using a backtracking strategy, which incurs extra computational cost. The advantage of scaled H-B is that it not only converges faster than the original version (**Lemma 3**), but, in practice, it is easily setup without additional cost by simply setting $\alpha^k = x^k$.

For the purposes of the experiments in the next section, we will evaluate three acceleration parameters; $\beta^{BA}$ defined by Eqn.(17), $\beta^{H-B}$, Eqn.(10), and $\beta^{FISTA}$, which is from the FISTA algorithm [12], and is defined as

$$\beta^k = \frac{t^k - 1}{t^{k+1}}, \tag{18}$$

where $t^{k+1} = \left[1 + \sqrt{1 + 4(t^k)^2}\right] \Big/ 2$. In fact, the FISTA algorithm can also be shown to be a special case of the scaled H-B method with $\alpha^k = 1/L$. For completeness, we will also evaluate the two reference algorithms, RL and BA.

## IV. EXPERIMENTS

In this section we demonstrate the performance of the scaled H-B method for different parameter choices on both synthetic and real 3D images. In all, we compare five algorithms; the RL algorithm (RL), the BA method, scaled H-B method with parameter $\beta^{BA}$ ($H-B\_\beta^{BA}$), scaled H-B method with parameter $\beta^{H-B}$ ($H-B\_\beta^{H-B}$) and scaled H-B method with parameter $\beta^{FISTA}$ ($H-B\_\beta^{FISTA}$). All algorithms are implemented in Matlab 7.12.0 and the BA method is the implementation in the deconvlucy function of the Image Processing MATLAB toolbox. The experiments are run on a Windows 7 64-bit machine with a processor Intel Core i7 2.80 Ghz and 6GB RAM.

### *A. Synthetic Image Results*

The synthetic dataset we use is called "hollow-bars" [20]. The 3D image, of size $256\times256\times128$ voxels, consists of six hollow bars arranged along a diagonal slice through a rectangular volume. As we know the synthetic reference image $x^*$, the performance measure is defined as the Kullback–Leibler (KL) divergence between $x^k$ and $x^*$

$$D_{KL}(x^k, x^*) = \sum_{i=1}^{n} \ln\left(\frac{x_i^k}{x_i^*}\right) x_i^k \tag{19}$$

To measure the speed-up factor, we let the RL algorithm run for 200 iterations to obtain a value for $D_{KL}(x^{200}, x^*)$. We then determine the number of iterations and time it takes for the other four algorithms to obtain a value smaller than $D_{KL}(x^{200}, x^*)$. We also run a stability measure to stop iterating if $\left|C(x^k) - C(x^{k-1})\right| \Big/ C(x^k) < T$, where $T$ = 0.00001 in all experiments.

Figure 1 shows the variation in KL divergence value versus iteration for both RL and scaled H-B with $\beta^{H-B}$,

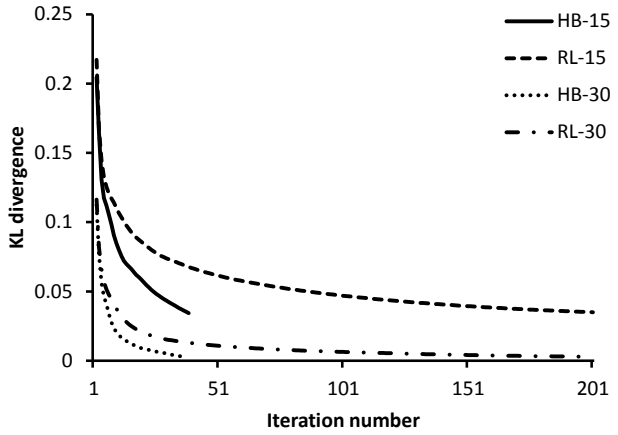


Fig. 1: Graph of KL divergence versus iteration number for both the RL algorithm and the scaled H-B with $\beta^{H-B}$ and for two noise levels (SNR =15 and 30) on the hollowbar image.

illustrating the faster convergence of the latter. Clearly, the RL algorithm takes the full 200 iterations to converge, whereas the scaled H-B converges to the same KL divergence value after only 38 iterations. The speed-up results for all five variants of the algorithm are summarized in Table I. The acceleration parameter $\beta^{BA}$ incurs extra computational cost to calculate, but reduces the number of iterations to 34. Both $\beta^{FISTA}$ and $\beta^{H-B}$ are trivial to calculate, but need slightly more iterations, i.e. 38. Overall, the three scaled H-B variants and the BA algorithm achieve similar speed-ups of around a factor of five.

Whilst the main focus of our work is on accelerating convergence, and not on improving image quality, we present here for completeness the restored images obtained with RL and scaled H-B, Fig. 2. The upper left is the original reference image, and the blurred and noisy version is upper right. The cause of the shadowing (or "ghost-bars") when blur and noise are added, is due to unfocused light from the adjacent horizontal bars occupying different z-planes. Comparison of the visual quality of the restored images shows that similar results are achieved with RL and scaled H-B. However, closer analysis shows that scaled H-B has reduced the shadowing of adjacent bars and has resolved the two edges of the in-focus bar marginally better.

| Algorithm | Iteration | Time (sec) | Speed-up |
|---|---|---|---|
| *RL* | 200 | 207 | 1.0 |
| *BA* | 29 | 38 | 5.4 |
| $H-B\_\beta^{BA}$ | 34 | 45 | 4.6 |
| $H-B\_\beta^{H-B}$ | 38 | 39 | 5.2 |
| $H-B\_\beta^{FISTA}$ | 38 | 40 | 5.2 |

TABLE I: Speed-up of scaled H-B on synthetic data.

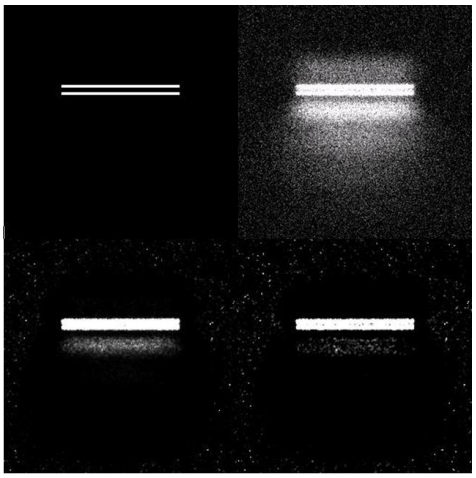

Fig. 2: One *z*-stack view of "hollow-bars". Original image (upper left); blurred noisy image (Upper right); restoration by RL (lower left) and scaled H-B (lower right).

In both cases the noise has been significantly reduced in the region around the bars, with some residual noise present in the outer background regions of the image. The latter is arguably more visible in the scaled H-B restored image.

To provide a quantitative evaluation of the restored images, the peak signal-to-noise ratio (PSNR) and Structural SIMilarity metrics (SSIM) [21] were used. TABLE II shows the scores for two noise levels, SNR =15 and 30, on the hollowbar image. Although the restoration scores decrease when the noise level increases, the various algorithms gave similar values, suggesting that acceleration does not come at a cost of sacrificing image restoration quality.

*B. Real Image Results*

The convergence rates of the algorithms were then evaluated using two real 3D microscopy datasets. The same test procedure used for the synthetic image was also used here, except that the performance measure in this case was $C(x)$ as we do not have a reference image to calculate $D_{KL}\left(x^{200}, x^{*}\right)$. In addition, we discovered that for the larger real image datasets, the 6 GB of RAM was insufficient to

| Algorithm | PSNR | SSIM | PSNR | SSIM |
|---|---|---|---|---|
| | SNR=15 | | SNR=30 | |
| *RL* | 19.59 | 0.803 | 18.80 | 0.478 |
| *BA* | 19.51 | 0.802 | 19.43 | 0.446 |
| $H-B\_\beta^{BA}$ | 19.42 | 0.807 | 18.87 | 0.461 |
| $H-B\_\beta^{H-B}$ | 19.59 | 0.818 | 19.12 | 0.448 |
| $H-B\_\beta^{FISTA}$ | 19.63 | 0.819 | 19.09 | 0.448 |

TABLE II: PSNR (dB) and SSIM scores of the various algorithms on the hollowbar image with SNR =15 and 30.

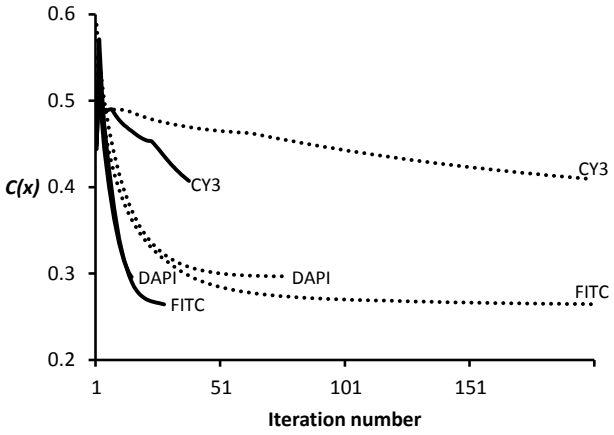


Fig. 3: Graph of $C(x)$ versus iteration number for both the RL algorithm (dotted line) and the scaled H-B with $\beta^{H-B}$ (solid line) for the C. Elegans embryo CY3, DAPI and FITC image channels.

store the variables required for the calculation of $\beta^{BA}$. This is a serious practical limitation of these algorithms, which we explore further in subsection *C*.

The first real dataset used is the image of a C. Elegans embryo [20], which consists of CY3, DAPI and FITC channels of size 672×712×104 voxels, Fig. 4 (left-hand column). Each channel contains different kinds of structures; extended objects (the chromosomes in the nuclei), filaments (the microtubules), and point-wise spots (a protein stained with CY3). The data was acquired with a100X, 1.4NA oil objective yielding an x-y image pixel size of 64.5 × 64.5 nm. The z-step size was 0.2 μm.

Figure 3 shows the convergence results for all three image channels of the C. Elegans embryo, with, and without, scaled H-B acceleration. Once again, the faster convergence of the scaled H-B is obvious, particularly for the FITC channel, even though it initially tends to oscillate for several iterations before starting to converge. A quantitative summary of the speed-up results for each channel image are shown in Table III. We observe a similar speed-up for the CY3 and DAPI image channels as for the "hollow-bars" image. In the DAPI channel case, the RL algorithm only runs for 78 iterations before meeting the stability criteria. The speed-up for the FITC channel is greater than for the others, a factor of seven compared to five.

| Algorithm | Iteration | Time(sec) | Speed-up |
|---|---|---|---|
| | **CY3** | | |
| *RL* | 200 | 1926 | 1.0 |
| $H-B\_\beta^{H-B}$ | 37 | 366 | 5.3 |
| $H-B\_\beta^{FISTA}$ | 37 | 368 | 5.2 |
| | **DAPI** | | |
| *RL* | 78 | 810 | 1.0 |
| $H-B\_\beta^{H-B}$ | 15 | 164 | 4.9 |
| $H-B\_\beta^{FISTA}$ | 15 | 161 | 5.0 |
| | **FITC** | | |
| *RL* | 200 | 2048 | 1.0 |
| $H-B\_\beta^{H-B}$ | 28 | 302 | 6.8 |
| $H-B\_\beta^{FISTA}$ | 28 | 303 | 6.8 |

TABLE III: Speed-up of scaled H-B on the C. Elegans image.

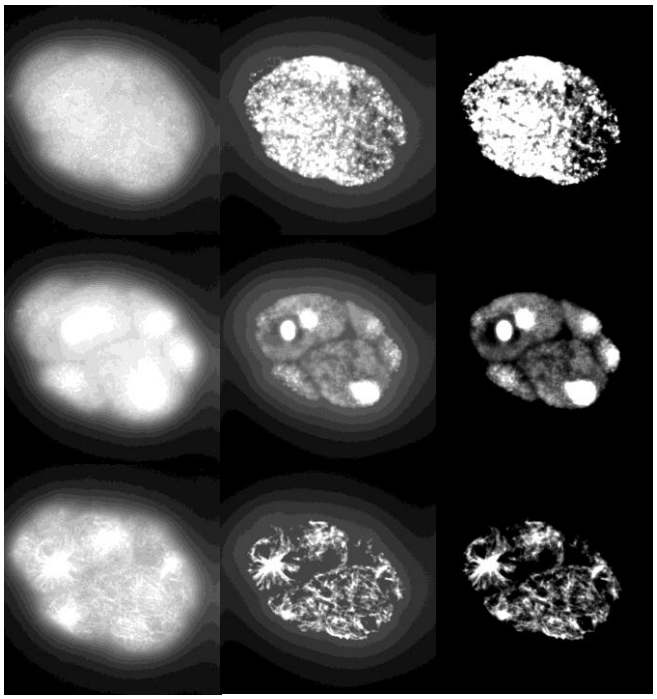

Fig. 4: One *z*-stack view of the CY3 (top), DAPI (middle) and FITC (bottom) channels of a C. Elegans embryo. (Left) Blurred noisy image; (Middle) Restoration by RL; (Right) Restoration by $H-B\_\beta^{H-B}$.

Figure 4 shows the restored images for all the channels. Once again, the scaled H-B appears to produce a better quality restored image in that it reduces the halo due to leakage of unfocused light from adjacent z-planes more than RL.

The third dataset is an image of a mouse kidney cell, Fig. 6, which consists of PFID_488 and PFID_560 channels of size 1004×1002×51 voxels. The PFID_488 channel, Fig. 6 (top-left), shows internal components and the PFID_560, Fig. 6 (bottom left), shows the outer component or membrane of the kidney. The data was acquired using a Spinning Disk confocal microscope with a 100X, 1.4NA oil objective, PlanApochromat. This yielded an x-y pixel size of 80 × 80 nm. The z-slice thicknesses were 0.13 μm and 0.2 μm for the PFID_488 and PFID_560 channels respectively.

Figure 5 shows the convergence results for both image channels of the mouse kidney, with, and without, scaled H-

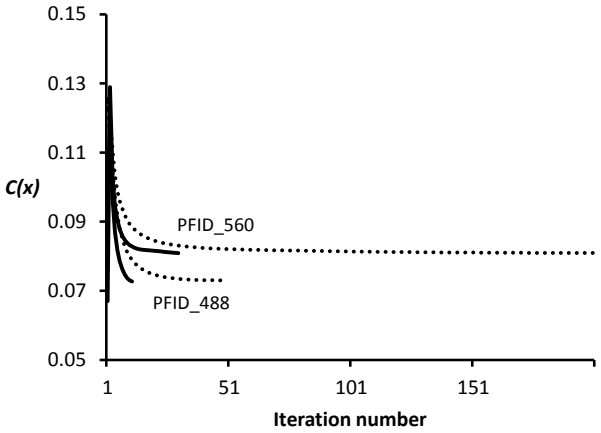


Fig. 5: Graph of $C(x)$ versus iteration number for both the RL algorithm (dotted line) and the scaled H-B with $\beta^{H\_B}$ (solid line) for the mouse kidney PFID_488 and PFID_560 image channels.

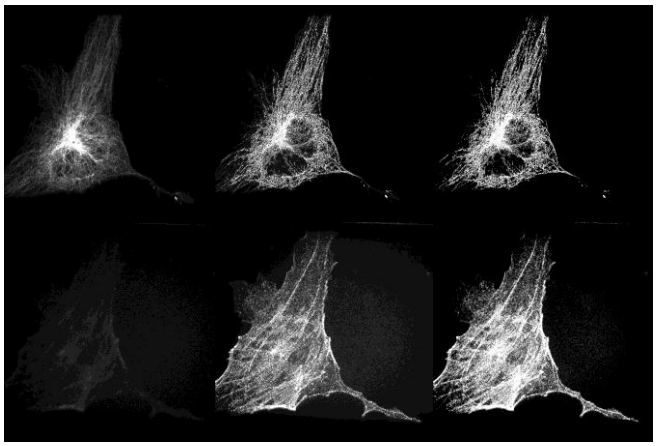

Fig. 6: One *z*- stack view of the PFID_488 (top) and PFID_560 (bottom) channels of mouse kidney. (Left) Blurred noisy image; (Middle) Restoration by RL; (Right) Restoration by $H\_B\_\beta^{H-B}$.

B acceleration. As before, the faster convergence of the scaled H-B is obvious. The timing results for both channels are summarised in Table IV. Once again, we observed a similar speed-up result for both image channels as for the "hollow-bars" image. For PFID_560, the RL algorithm only runs for fifty iterations before it meets the stability criteria.

The restored images for both channels are also shown in Fig. 6. In this case, for PFID_488, the degradation in image quality in the original image was not as severe as for the other datasets. Nonetheless, the image quality has improved with fine internal components being resolved. The contrast for the H-B image is marginally better than for the RL image. For PFID_560, the original image has very poor contrast. Both restored images are significant improvements on the original, with the H-B image again having marginally better contrast than the RL image.

### *C. Memory analysis*

The peak memory usages of RL, BA and scaled HB are shown in Table V for all three image datasets. It is clear that the memory usage for RL and $H-B\_\beta^{H-B}$ is roughly the same, whereas it is greater for BA as indicated previously. For the hollowbar image, roughly 35% extra memory is required for BA, whilst for the real images the system ran out of memory.

| Algorithm | Iteration | Time(sec) | Speed-up |
|---|---|---|---|
| | **PFID_488** | | |
| *RL* | 200 | 2951 | 1.0 |
| $H-B\_\beta^{H-B}$ | 30 | 459 | 6.4 |
| $H-B\_\beta^{FISTA}$ | 29 | 455 | 6.5 |
| | **PFID_560** | | |
| *RL* | 50 | 741 | 1.0 |
| $H-B\_\beta^{H-B}$ | 11 | 172 | 4.3 |
| $H-B\_\beta^{FISTA}$ | 10 | 156 | 4.7 |

TABLE IV: Speed-up of scaled H-B on the mouse kidney image.

## V. CONCLUSIONS

In summary, the main contribution of this paper is to propose a novel scaled H-B method for accelerating the RL algorithm. We show that the method includes the BA method as a special case, and provide a proof that it has a convergence rate of $O\left(k^{-2}\right)$. Experiments on both synthetic and real images, demonstrate a speed-up factor of around five on average over the standard RL algorithm. In addition, we would argue that there is also an associated marginal improvement in the quality of the restored image obtained using scaled H-B. In the future, we'd like to extend the analysis here to include a regularization term.

ACKNOWLEDGEMENTS

The authors would like to acknowledge funding from Invest NI grant ST285 and Andor Technology.

| | Image size (voxels) | RL | $H-B\_\beta^{H-B}$ | BA |
|---|---|---|---|---|
| **Hollow-bar** | 256×256×12 | 887 | 829 | 1152 |
| **C. Elegans embryo CY3** | 672×712×104 | 4729 | 4728 | Out of memory |
| **Mouse kidney PFID_488** | 1004×1002×51 | 4470 | 4644 | Out of memory |

TABLE V: Peak memory usage, in MB, of RL, BA and scaled H-B.

## REFERENCES


[1] W. H. Richardson, "Bayesian-based iterative method of image restoration," J. Opt. Soc. Am. A, vol. 62, no. 1, pp. 55–59, 1972.

[2] L. B. Lucy, "An iterative technique for the rectification of observed distributions," Astron.J, vol.79, no.6, pp.745–754, 1974.

[3] M. Bertero, P. Boccacci, G. Desidera, and G. Vicidomini, "Image deblurring with Poisson data: from cells to galaxies," Inverse Problems, 25(12):123006, 2009.

[4] G.M.P. van Kempen and L.J. van Vliet, "Background estimation in non linear image restoration," Journal of Optical Society of America A, vol. 17, no. 3, pp. 425-433, Mar. 2000.

[5] T.J. Holmes and Y.H. Liu "Acceleration of maximum-likelihood image restoration for fluorescence microscopy," J. Opt. Soc. Am. A-8 893–907, 1991.

[6] H. Lant´eri, M. Roche, O. Cuevas and C. Aime "A general method to devise maximum-likelihood signal restoration multiplicative algorithms with nonnegativity constraints," Signal Process., 81, 945–74, 2001.

[7] D. S. C. Biggs and M. Andrews, "Conjugate gradient acceleration of maximum-likelihood image restoration," Electron. Lett. 1995.

[8] D.S.C. Biggs and M.Andrews, "Acceleration of iterative image restoration algorithms," Appl. Opt.36, 1766–75, 1997.

[9] S. Bonettini, R. Zanella and L. Zanni, "A scaled gradient projection method for constrained image deblurring," Inverse Problems, 25, 015002, 2009.

[10] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$", Soviet Math. Doklady, vol. 27, no. 2, pp.372 -376, 1983.

[11] J. Bioucas-Das and M. Figueiredo, "A new twist: two step iterative shrinkage/thresholding algorithms for image restoration," IEEE Trans. On Image Proc., vol. 16, no. 12, pp. 2992–3004, Dec 2007.

[12] A. Beck and M. Teboulle, "A fast iterative shrinkage thresholding algorithm for linear inverse problems," SIAM Journal on Imaging Sciences, vol. 2, no. 1, pp. 183–202, 2009.

[13] Wangmeng Zuo and Zhouchen Lin, "A Generalized Accelerated Proximal Gradient Approach for Total-Variation-Based Image Restoration," IEEE Trans. Image Process., vol. 20, no.10, pp. 2748–2759, Oct. 2011.

[14] H. Wang and P. C. Miller, "Scaled Heavy-Ball Acceleration of the Richardson-Lucy Algorithm", in *Proc. IEEE Int. Symp. Biomed. Imag.*, 2012, pp.1743-1746.

[15] B. T. Polyak, "Introduction to Optimization," Software. Inc., N.Y., 1987.

[16] Dimitri P. Bertsekas, "Convex Optimization Theory, Supplementary Chapter 6 on Convex Optimization Algorithms," Athena Scientific, 2009.

[17] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," IEEE Trans. Image Process., vol. 19, no.12, pp.3133–3145, 2010.

[18] F.-X. Dup´e, J. Fadili, J.-L. Starck, "A proximal iteration for deconvolving Poisson images using sparse representations", IEEE Trans. on Image Proc., vol. 18, pp. 310–321, 2009.

[19] Z. T. Harmany, R. F. Marcia and R. M. Willett, "This is SPIRAL-TAP: Sparse Poisson Intensity Reconstruction Algorithms — Theory and Practice", IEEE Transactions on Image Processing, vol. 21, no. 3, 1084–1096, Mar. 2012.

[20] http://bigwww.epfl.ch/deconvolution/?p=

[21] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004.